This is the accepted version of an article that has been published in the ICE/ITMC 2021 proceedings.
Minor changes, primarily to formatting, were made to this version by the publisher prior to publication.
The final version of record is available at
https://doi.org/10.1109/ICE/ITMC52061.2021.9570250

Master Data Taxonomy - A systematic approach to assess and migrate master data

Matthias Knapp
KDQ Knapp Data Quality
Aachen, Germany
matthias.knapp@kdq.de

Theo Lutz
Department of Business and Industrial Engineering
Offenburg University of Applied Sciences
Offenburg, Germany
Theo.Lutz@hs-offenburg.de

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

978-1-7281-3401-7/18/$31.00 ©2021 IEEE

Master Data Taxonomy - A systematic approach to assess and migrate master data

Matthias Knapp
KDQ Knapp Data Quality
Aachen, Germany
matthias.knapp@kdq.de

Theo Lutz
Department of Business and Industrial Engineering
Offenburg University of Applied Sciences
Offenburg, Germany
Theo.Lutz@hs-offenburg.de

***Abstract*— This paper describes a taxonomy which allows to assess and compare different implementations of master data objects. A systematic breakdown of core entities provides a framework to tell apart four subdividing categories of master data objects: independent and dependent objects, relational objects, and reference objects that serve to attribute information. This supports the preparation of data migrations from one system to another.**

Keywords*— *master data, reference data, data model, data architecture, taxonomy, data migration, business applications

I. Introduction

Challenges in Master Data Migrations

Apart from a functional model, the design of business applications is always based on a data model which represents the required data objects and their relationships. The exact understanding and specification of this data model and its elements is essential for various issues, e.g., for the transformation of data from one data architecture to another (data migration) as well as for consolidating data from several sources (data integration).

In data migration projects, the predominant approach to transferring data from legacy systems is still "trial and error." The providers of the new system often provide target data structures as a template which are usually not matched against the available data and the respective data model. In the course of the project, tests are often performed with test data that already corresponds to the target structure but do not reflect the structure of the legacy system. Therefore, incompatibilities are first discovered at a late stage. In the best case, they lead to additional effort for troubleshooting, but always bear the risk that fundamental conceptual changes must be made and complex mappings are inevitable. In the worst case, the project is at risk to fail due to enormous additional effort.

That said, comparing and transferring master data among different data models is the central challenge in migrating business applications. The cause is a large and obscure variety of data models being used in different applications [1]. Not even the fundamental data objects such as customer, vendor or item master are implemented following a universal or standardized schema.

For instance, the modeling of the most fundamental information about a company in a customer record shows significant differences between the various implementations. On the one hand, this concerns the way recurrently used values for attributes are being provided, such as countries. Here, applications may allow either free text or a selection from a defined list of values. On the other hand, differences in data models may also concern the relationship between information, e.g. the relationship between company name and multiple addresses such as billing and shipping addresses. Multiple addresses are often provided in the same object (or table, respectively), whereas other applications define dependent address objects that can be instantiated multiple times.

It is obvious that these differences in the information and data models of different applications have a significant impact on usability and, in the case of data migration, on the need to transform data. In some cases, a substantially different data model may hinder to completely transfer data from one application to another. Take again addresses as an example: If the source system comprises three or more addresses in a 1:n relation but the data model of the target system allows only two addresses per customer within the customer table, the two data models are not compatible.

Research question

To document and compare the design of different data models, an application-neutral description of their elements is required. Based both on literature research and an analysis and comparison of different ERP implementations available on the market, following research questions need to be answered:

- *What are properties of master data objects that help to detect incompatibilities between data models of different IT systems?*
- *What are relevant characteristics of these properties?*

II. Existing Theories and previous Work

A commonly accepted definition of data types is the distinction between transactional and master data. While the first type contains data representing information needed to execute daily business processes (e.g., orders) [2], it is always related to the latter. After finishing a process, its transactional data is usually not needed any longer except for administrative purposes (e.g., documentation). Although in practice there is a common understanding which data objects belong to the master data and which to transactional data, formal definitions often remain vague.

The research process on existing definitions that may build a foundation for solving our research question has been carried by first using a search for the key words “master data, definition” and their respective German translations “Stammdaten, Definition” in the “bucket of eight”, Google Scholar and EBSCOhost. We then did a backward search based on the citations of the articles found in step one and a forward search to identify further relevant articles. Our research led to an extensive list of sources. The most relevant papers are listed in TABLE I.

While the research did not reveal an applicable definition, a few common characteristics can be identified. Master data is frequently considered to represent "core entities". Master data represents objects shared among departments and remaining unchanged over time but is repeatedly used.

The available sources usually support their definitions of master data with examples, namely items, bills of material, prices (in the domain of item data) or vendors, addresses, bank accounts and prices (in the domain of business partner data). Obviously, these objects differ in their information content and in their relationship to each other. Some, like prices, may even belong to different data domains.

TABLE I. COMMONLY STATED CHARACTERISTICS OF MASTER DATA

source	master data is typically ...			
	... core data	*... unchanged*	*... repeatedly used*	*... used across departments*
LEGNER AND OTTO [3]	X	X	X	X
SCHEMM [4]	X	X		
PIRO AND GEBAUER [5]	X		X	X
HAUG AND STENTOFT [6]	X	X	X	
BERSON AND DUBOV [7]	X		X	X
WEGENER [8]			X	X
SCHEMM ET AL. [9]	X	X	X	
LOGAN ET AL. [10]	X	X	X	X
MERTENS ET AL. [11]	X			
LOOS [12]	X	X	X	
SCHWARZER A. ROGGE [13]		X		
BENZ A. HÖFLINGER [14]		X	X	
HILDEBRAND ET AL. [15]		X		
BERNHARDT A. LIEBING [16]		X	X	
ENNEMANN A. RÜCKERT [17]		X	X	X
HILDEBRAND [18]		X		
OTTO A. LEGNER [19]	X			
FAN ET AL. [20]	X			X
SCHEUCH ET AL. [21]	X			
KEUPER ET AL. [22]	X			
DREIBELBIS ET AL. [23]	X		X	X
OFNER ET AL. [24]	X	X		X
LOSHIN[25]	X	X		X

Three of the publications will be examined in more detail as examples

LEGNER AND OTTO emphasize that master data as essential core data of a company is used by different functional areas. In addition to general basic data, it also models process-specific information. By integrating the entire information of an object in one master record, redundant data storage can be avoided [3].

SCHEMM addresses the fact that master data includes a variety of attributes and points out that these are broken down into a variety of different views for functional or organizational grouping [4].

HILDEBRAND also differentiates master data according to its use and distinguishes global master data, which is used throughout the entire company, from process-specific or local master data [18].

However, these approaches do not take into account how the various information components are modeled or how the information relates to each other. The general definition of master data is not further subdivided. A differentiation of elements, which in particular also considers the variety of possible implementations, is not made.

The only special kind of master data mentioned is the so-called “reference data”. According to [4], reference data comprises value lists to classify other master data objects, is often managed externally and is typically defined by standardization initiatives. In contrast, company-specific, internally managed value lists, e.g. material groups or sales territories, remain unmentioned in literature so far.

By introducing a concept of “segments” and “attributes”, [21] gives a brief idea how to structure the various elements of master data. Still, a comprehensive and detailed taxonomy of those elements is missing. This paper therefore aims at developing such a taxonomy.

III. METHODOLOGY

Definition and approach

Taxonomies are widely used in different scientific fields such as biology, engineering and computer science. They seek to order entities within a given domain such as real-world objects, concepts or terminology. The term “taxonomy” can refer to the process of classifying entities or concepts as well as the result of such a process which is as a scheme of classification that is often hierarchical [26]. It can also be used to describe a set of dimensions with each having two or more characteristics and each object that should be categorized fulfills exactly one characteristic in each dimension [27]. Based on this understanding, Nickerson et al. propose a method for developing taxonomies in information systems [28] (Fig. 1) which shall build the methodological foundation of this paper.

Requirements

Nickerson et al. define five quality requirements a taxonomy should meet in order to be useful [27, 28], which have been picked up by various other authors:

1. Conciseness: The number of dimensions and characteristics in each dimension should be limited to make it easy to understand.

2. Inclusiveness: The number of dimensions and characteristics should be high enough for taxonomy "to be of interest" as the authors state. We interpret this as having enough dimensions to sufficiently characterize relevant objects within the objective of the taxonomy.
3. Comprehensiveness: The taxonomy should be able to characterize all current objects within the domain that the taxonomy claims to describe.
4. Extendibility: It should be feasible to extend the taxonomy if necessary, e. g. when new types of objects appear
5. Explanatory: The taxonomy should describe the type of object, meaning their nature but not their content.

We add following requirements that need to be fulfilled to support the objective of this paper:

a) The taxonomy needs to support the identification of incompatibility between data models.
b) Data objects need to be characterized in a way that allows to identify the order in which data needs to be migrated.
c) The characterization therefore must be independent from specific IT systems.

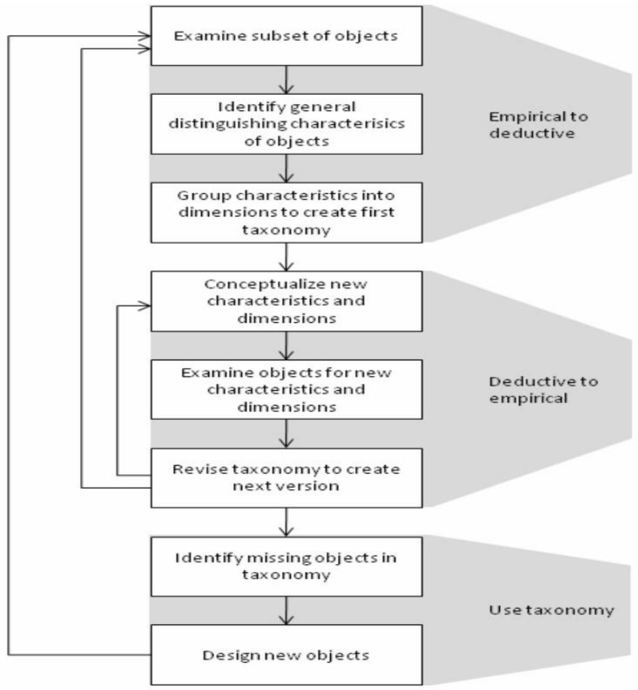

Fig. 1. Taxonomy development method [28]

Application of the method

To start with, we defined a subset of master data models to be analyzed. We limited our examination to central master data objects in the ERP applications SAP ECC and proAlpha. This subset contained data from the domains "customer" and "material" (i.e. parts), including "bill of material" (BOM, including BOM headers and BOM lines), data related to both material and customer ("info records"), and "countries".

We then analyzed similarities and differences in the implementation of the particular data model. Significant differences were observed, above all, in the allocation of information to different objects and in the table relations of the respective data model. For instance, in SAP ECC the customer object encompasses address information within the general section, whereas in proAlpha an address (of any kind of business partner) is always handled as a separate object.

In several iterations, these findings then have been generalized and grouped into two characteristics relevant for data migration. We applied the characteristics to various other data objects within SAP ECC and proAlpha and finally cross-checked or findings against the implementation of the ERP applications Infor LN and PSI PENTA.

IV. Foundation of the Taxonomy of Master Data

A systematic and concise distinction between reference data and master data, as well as their differentiation from transaction data, forms the core of a taxonomy of master data objects. These definitions must meet the formal requirements described above and thus goes beyond the definitions found in the literature.

Business objects are usually characterized with the help of one or more attributes. The attributes are specified by entering values, e.g. customer name as free text, or by assigning values, e.g. country from a country list.

Based on this, a first distinction of data objects can be made regarding their purpose. Data objects either provide a predefined set of attribute values that can be used by other objects, or they define which attributes are used to characterize business objects.

The second distinction concerns the way in which data objects can be created. On the one hand, there are data objects that are created by business processes in the course of carrying out business activities, such as the creation of customers or material documents, and on the other hand, there are data objects that are never created by the execution of a business process. They are initially created and only very rarely adjusted. This can happen, for example, in the case of structural changes such as the renaming of countries or adjustments to charts of accounts. The frequency of change of data objects, however, is not a sufficiently distinguishing feature, since customs tariff numbers, for example, are updated annually, while address data of customers usually change less frequently and may even never change. Data objects such as receipts, which are commonly referred to as transaction data, never change.

TABLE II. Basic Data Object Morphology

Features	Characteristics	
Purpose	Define which attributes can be used to characterize business objects	Provide predefined set of attribute values
Creation	Initially created	Regularly created
Recurring use	Yes	No

The third differentiating feature relates to frequency of use. Data objects are either used repeatedly in business processes that are carried out on a regular basis, or they are only used for documentary or analytical purposes after a business process has been completed. The three features and their characteristics are displayed in TABLE II.

With the help of these characteristics, reference data can be typed as such data objects, which provide attribute values and are initially created. They are intended to be used repeatedly.

Master data are data objects that define which attributes can be used to characterize business objects. They are created regularly by business processes and they are also intended to be used repeatedly.

Transaction data are data objects that define the attributes by which business objects are characterized. They are regularly created by business processes. However, in contrast to master data and reference data, they are not used again after completion of a business process, but serve then only documentary and analytical purposes.

Transaction data will not be considered in the remainder of this paper. Within the two other types of data objects, however, further distinctions must be made, which we explain below.

V. Elements of the Taxonomy of Master Data

Reference Data Objects

Attributes characterize entities and thus classify master data objects. Values of those attributes may represent different types of data, e.g. numeric, string, Boolean or date. In any case, the specification of a value may result from a user's input or selection, or an algorithmic calculation within the application. If selected from a list, these options are typically either enumerated within the program code or stored in a separate lookup table for reference.

These reference data objects typically provide a curated set of values that can be selected and repeatedly used to characterize and classify other data objects [29, 30]. They do not change as a result of executing internal business processes.

To systematically tell apart different implementations, the following criteria can be used: The existence of a primary key, the use of foreign keys and the number of attributes covered by the reference object (TABLE III.).

TABLE III. Reference Data Object Morphology

Features	**Characteristics**	
Existence of a Primary Key	no PK	PK
Use of Foreign Keys	no FK	FK
Number of Attributes	1	>1

This leads us to distinguish four types of reference data (see TABLE IV.).

Trivial Reference Data Objects are just a one-column list of values, such as a plain list of colors or basic materials without any primary key or code being used.

Simple Reference Data Objects also contain no more than one attribute, but add a key column often containing a code or acronym representing the full description of the value [30]. For instance, a list of countries may be represented by codes according to ISO 3166, which sometimes even replace the value itself in display and print routines.

Regardless of their implementation, Trivial and Simple Reference Data Objects define a very basic set of master data that is necessary and useful for working with an IT system but provide little to no value if not used in master data objects.

Extended Reference Data Objects extend Simple Reference Data Objects by appending more columns with additional information. Therefore, Extended Reference Data Objects should be implemented as tables, although it cannot be ruled out that they are hard-coded in poorly designed IT systems. An example might be a Reference Data Object that provides a list of countries with their respective country phone code, ZIP and VAT number format.

Complex Reference Data Objects are similar to the latter but may include information provided by another reference data object, either directly or using a foreign key. So, for example, a country list can be enhanced by a language column, which is referring to a separate reference data object. Complex Reference Data Objects are always implemented as tables. The resulting dependencies lead to increased complexity. An instance cannot be created without prior existence of the lookup reference because the values of its attributes are provided by other objects.

TABLE IV. Types of Reference Data Objects

	1 attribute		**>1 attribute**	
	no PK	***PK***	***PK***	***no PK***
no FK	Trivial	Simple	Extended	-
FK	-	-	Complex	-

Any other combination is leads to technically inadequate implementations. If the reference data object just contains one attribute, it does not make sense to use a foreign key in it. And if the reference data contains more than one attribute, it is inevitable to define a primary key.

There are two types origin for reference data. While some of the reference data is originated from the outside and mostly standardized, such as the previously mentioned country codes, a vast amount of reference data is defined by and within the company, like product groups or account plans. Although this distinction is not necessary to fulfill one of the requirements stated above and therefore is not required in our taxonomy, it can be used to extend the taxonomy and further distinguish different types of reference data.

However, any type of reference data object has no worth from itself but only provides business value indirectly, being used to populate fields of other data objects. Still, reference data can contribute significantly to the design of those objects by providing parts of their identifier. For instance, they can form a composite key, as material groups might be part of a speaking article number.

Master Data Objects

The type-forming characteristic of master data objects is, as described above, to define a framework for describing entities through attribute structures, unlike reference data objects that just provide defined values -that is, content- for attributing other objects. Within this definition, the Master Data Objects have further diversities and can be divided into different categories. An appropriate criterion for this is the relationship or dependency between the objects. There are Master Data Objects that do not depend on any object at all or only on Reference Data Objects that provide values for their attributes. Other Master Data Objects depend on exactly one single Master Data Object while there is a third group of Master data Objects that depends on multiple other Master Data Objects (TABLE V.). The assessment to which group a Master Data Object belongs is based on the number and type of foreign keys. It turns out that this feature is the only one required to further distinguish different types of Master Data and fulfill the requirements stated above.

TABLE V. MASTER DATA OBJECT CLASSIFICATION

Features	**Characteristics**		
Dependency on other objects	None or Referential only	One non-referential	Multiple non-referential

a) Primary Master Data Objects

Primary Master Data Objects form the core of master data objects. They may contain foreign keys of Reference Data Objects to define values of their attributes, but they do not contain foreign keys of other types of master data objects (see Fig. 2). Typical examples are general material data (tables “MARA” in SAP ECC, “Item” in Infor LN, “Teile” in proAlpha, “S_Teile” in PENTA) or general supplier data (tables “LFA1” in SAP ECC, “Business Partners” in Infor LN, “S_Lieferant” in proAlpha, “PLIF” in PENTA). The amount of the data stored in such an object is independent from this definition. It may vary from just holding simple identifiers to holding any kind of information, including addresses, bank accounts or communication details such as telephone numbers and email addresses. In extreme cases, Primary Master Data Objects contain only their identifier.

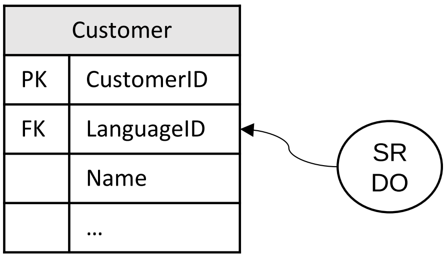

Fig. 2. Example for Primary Master Data Object with attribute from Simple Reference Data Object (SRDO)

As data models differ between IT systems, it is not possible to derive a content-based list of master data objects that represent Primary Master Data Objects. For example, quite counter-intuitively but in line with our definition, in SAP the BOM header (table STKO) represents a Primary Master Data Object, since no foreign keys of other master data tables besides reference data are included. The link to the MARA table is established via a separate table. In other systems, however, the BOM header may be attached directly to the material via a foreign key in the corresponding table. In such a case, we would call the BOM header a Dependent Master Data Object, which we define thereafter.

b) Dependent Master Data Objects

In addition to foreign keys of Reference Data Objects, Dependent Master Data Objects contain exactly one foreign key that does not belong to Reference Master Data Objects.

Such a key could be part of a Primary Master Data Object or of another Dependent Master Data Object. Dependent Master Data Objects extend a Primary Master Data Object directly or indirectly by additional characteristics (see Fig. 3). An example in SAP ERP would be plant data (table MARC) or company code-specific vendor data (table LFB1).

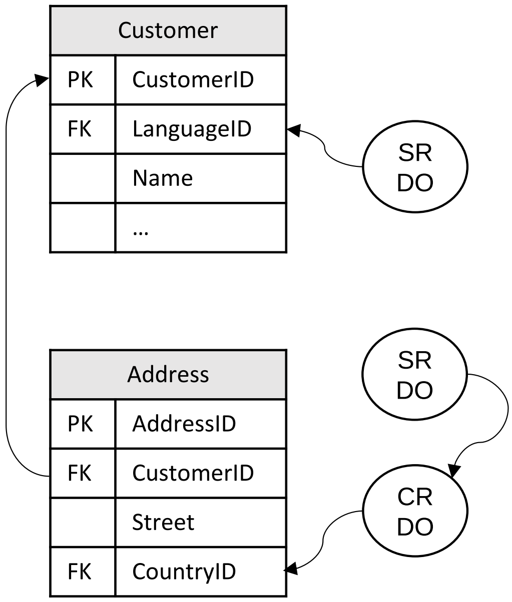

Fig. 3. Example for Primary and Dependent Master Data Object with attributes from Simple Reference Data (SRDO) and Complex Reference Data Object (CRDO)

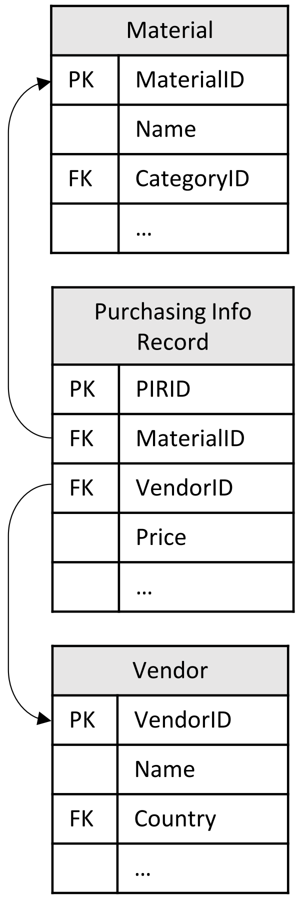

Fig. 4. Example for Relational Master Data Object connecting two Primary Master Data Objects

c) Relational Master Data Objects

In many cases, there are extended relations linking different master data objects. For example, items can be linked to vendors via the purchasing info record. Such relations are often described in more detail using separate master data objects, such as a purchasing info record. We call these objects Relational Master Data Objects. They contain two or more foreign keys belonging to a Primary or Dependent Master Data Object (see Fig. 4). Additionally, they can also contain foreign keys of Reference Master Data Objects like all other objects. As an example, SAP ECC stores the information related to both item (material) and vendor in the purchase info records' table EINA.

d) Graph of master data objects and the order of master data objects

Since Dependent Master Data Objects can also contain a foreign key of another Dependent Master Data Object instead of a foreign key of a Primary Master Data Object, chains of master data objects can be identified in data models. Relational Master Data Objects link at least two of such chains and therefore create a graph of master data objects (see Fig. 5). By convention, we define that there is only one edge between two master data objects that is directed from the object that holds a foreign key to the object where the key originated. The graph is therefore oriented.

One important measure in such a graph is the distance of a master data object to other data objects which we define as the number of steps between these two objects. The length of the path to the most distant Reference Data Object defines the order of a master data object. Simple Reference Data Objects are always of order zero. Extended Master Data Objects can be of order zero or higher. Primary Master Data Objects that only use attributes of Simple Reference Data Object are of order 1. Dependent Master Data Objects that are directly attached to exactly one Primary Master Data Object or one Dependent Master Data Object are of one order higher than their successor.

VI. APPLICATION OF THE TAXONOMY

Several examples illustrate the practical benefits of the taxonomy presented here, all taken from master data migration projects.

In the first example, objects within the domain "article master" of two ERP applications have been typified and compared. In both applications there was one master data object each for articles, bills of material and routings. The article master was a primary master data object in both applications. However, the relationship of the dependent objects revealed a significant difference. In application A, several routings were second order Dependent Master Data Objects. Several BOMs were assigned to them as third order Dependent Master Data Objects. The order of Dependent Master Data Objects in application B was inverse: several routings were directly assigned to the article as second order Dependent Master Data Objects, and several alternative BOMs were dependent from them. So, the classification of the objects immediately showed that a mapping of the objects and therefore a migration was not possible without a complex transformation. In this case new article variants had to be created for thousands of articles. If this additional effort had been known beforehand, another application would have been selected.

A second example is also taken from the article master domain. We found that within its "article" table an article directly referenced another article as predecessor. Hence this master data object served simultaneously as Primary and Dependent Master Data Object. Using our taxonomy exposed self-referencing edges and therefore allowed to easily unveil the need to split batches of article data for sequential migration.

In a third example, the tables, respectively data objects, of a source system have been analyzed and their relationship to each other was documented. From the order of the dependent data objects, described by their ordinal numbers, the sequence of data transfer into the identically structured target system could be derived without any problems.

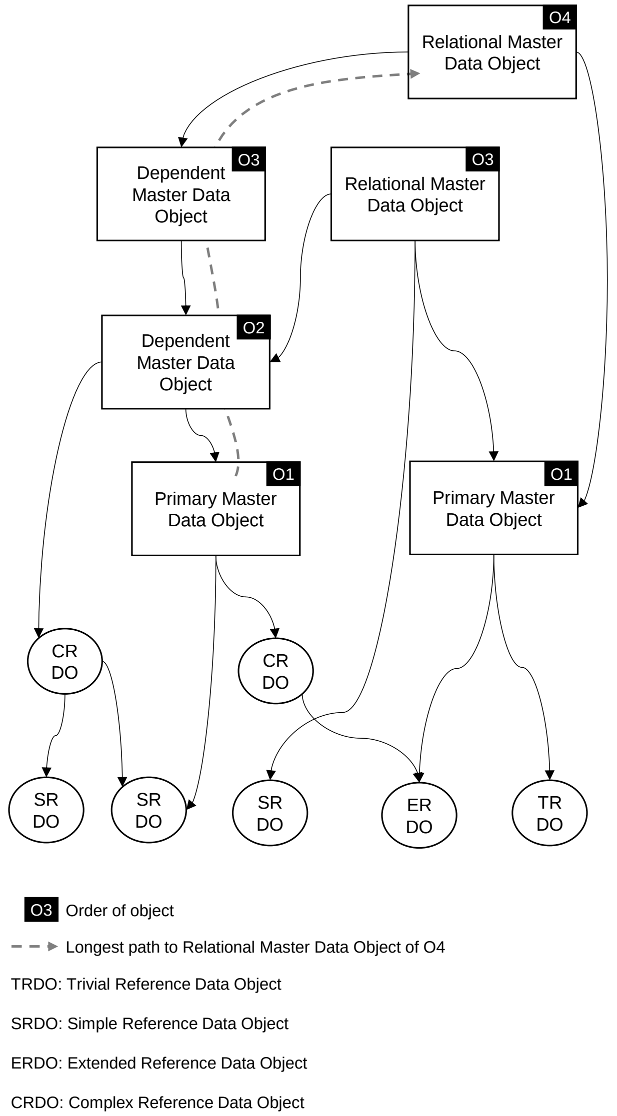

Fig. 5. Graph of master data objects

VII. CONCLUSION AND OUTLOOK

With the model described in section IV, we have a simple but powerful taxonomy for classifying elements in data models. This serves to identify differences in data models of two IT systems which is especially important for the preparation of data migrations from one system to another.

The taxonomy described also covers other concepts that are part of master data, e.g. translation tables. Those refer to a superior master data object and have an attribute "language" specified by reference data. Thus, they can be clearly identified as dependent master data. In contrast, naming catalogs, which are often used for standardizing product descriptions, would be considered as reference data objects, since they provide attribute values, are populated initially, and are used on a regular basis.

Ongoing studies will show if the results shown are also transferable to other applications such as CRM, PLM, and others. In the area of self-referencing objects, further analysis will be useful to uncover any special cases and to comprehensively validate the feasibility of the method presented.

Additionally, in the process of further research, techniques can be developed to automatically determine not only metadata in unknown database architectures, but above all the underlying structure and dependencies among master data objects. Based on existing object references, a directed graph can thus be determined, which can be used to significantly facilitate both the quality-assured preparation and the transfer of data in migration projects.

REFERENCES

[1] J. Lu and I. Holubová, "Multi-model Databases: A New Journey to Handle the Variety of Data," *ACM Computing Surveys*, vol. 52, no. 3, pp. 1–38, 2019, doi: 10.1145/3323214.

[2] A. Schmidt, *Entwicklung einer Methode zur Stammdatenintegration*. Diss. Nr. 3848 Wirtschaftswiss. St. Gallen. Berlin: Logos, 2010. Accessed: Mar. 21 2021.

[3] C. Legner and B. Otto, "Stammdaten-Management," *Das Wirtschaftsstudium (WISU)*, vol. 236, no. 4, pp. 562–568, 2007. [Online]. Available: www.alexandria.unisg.ch/67534/

[4] J. W. Schemm, "Architektur für das Stammdatenmanagement," in *Business Engineering, Zwischenbetriebliches Stammdatenmanagement: Lösungen für die Datensynchronisation zwischen Handel und Konsumgüterindustrie*, J. W. Schemm, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 79–214.

[5] A. Piro and M. Gebauer, "Definition von Datenarten zur konsistenten Kommunikation im Unternehmen," in *Daten- und Informationsqualität: Auf dem Weg zur Information Excellence*, K. Hildebrand, M. Gebauer, H. Hinrichs, and M. Mielke, Eds., 4th ed., Wiesbaden: Springer Vieweg, 2018, pp. 141–154.

[6] A. Haug and J. Stentoft Arlbjørn, "Barriers to master data quality," *Journal of Enterprise Information Management*, vol. 24, no. 3, pp. 288–303, 2011, doi: 10.1108/17410391111122862.

[7] A. Berson and L. Dubov, *Master data management and data governance*, 2nd ed. New York: McGraw-Hill, 2010. [Online]. Available: http://lib.myilibrary.com/Open.aspx?id=302153

[8] H. Wegener, "Metadaten, Referenzdaten, Stammdaten," in *Active Enterprise Intelligence™*, J. Töpfer and R. Winter, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 179–199.

[9] J. W. Schemm, C. Legner, and H. Österle, "Global Data Synchronization: Lösungsansatz für das überbetriebliche Produktstammdatenmanagement zwischen Konsumgüterindustrie und Handel?," in *Wertschöpfungsnetzwerke*, J. Becker, R. Knackstedt, and D. Pfeiffer, Eds., Heidelberg: Physica-Verlag HD, 2008, pp. 173–192.

[10] D. Logan, J. Radcliffe, A. White, and D. Newman, *Mastering Master Data Management.* [Online]. Available: https://www.gartner.com/en/documents/488207/mastering-master-data-management

[11] P. Mertens *et al., Eds., Lexikon der Wirtschaftsinformatik*. Berlin, Heidelberg, s.l.: Springer Berlin Heidelberg, 1997.

[12] P. Loos, "Grunddatenverwaltung und Betriebsdatenerfassung als Basis der Produktionsplanung und –steuerung," in *Controlling, Einführung in das Produktionscontrolling*, H. Corsten, Ed., München: Vahlen, 1999.

[13] J. Schwarzer and R. Rogge, *Wirtschaftsinformatik: Nachschlagewerk für Studium und Praxis,* 1st ed. Wiesbaden: Gabler, 2006.

[14] J. Benz and M. Höflinger, "Stammdaten," in *Logistikprozesse mit SAP R/3®*, J. Benz and M. Höflinger, Eds., Wiesbaden: Vieweg+Teubner Verlag, 2005, pp. 59–148.

[15] K. Hildebrand, M. Gebauer, H. Hinrichs, and M. Mielke, Eds., *Daten- und Informationsqualität: Auf dem Weg zur Information Excellence*, 4th ed. Wiesbaden: Springer Vieweg, 2018.

[16] R. Bernhardt and P. A. Liebing, "Stammdaten," in *Praxis der EDV - Einführung*, R. Bernhardt and P. A. Liebing, Eds., Heidelberg: Physica-Verlag HD, 1983, pp. 45–63.

[17] M. Ennemann and J. Rückert, "Mit validen Stammdaten in die Zukunft," *Control Manag Rev*, vol. 60, no. 3, pp. 24–33, 2016, doi: 10.1007/s12176-016-0036-5.

[18] K. Hildebrand, "Management von Logistik-Stammdaten in SAP," *HMD*, vol. 55, no. 1, pp. 76–90, 2018, doi: 10.1365/s40702-017-0374-6.

[19] B. Otto and C. Legner, "Master Data erfolgreich managen," *Control Manag Rev*, vol. 60, no. 3, pp. 6–17, 2016, doi: 10.1007/s12176-016-0038-3.

[20] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu, "Towards certain fixes with editing rules and master data," *Proc. VLDB Endow.*, vol. 3, 1-2, pp. 173–184, 2010, doi: 10.14778/1920841.1920867.

[21] R. Scheuch, T. Gansor, and C. Ziller, *Master Data Management: Strategie, Organisation, Architektur,* 1st ed. Heidelberg: Dpunkt, 2012.

[22] F. Keuper, K. Hamidian, E. Verwaayen, and T. Kalinowski, *transformIT: Optimale Geschäftsprozesse*

durch eine transformierende IT. Wiesbaden: Gabler, 2010.
[23] Dreibelbis, A., Hechler, E., Milman, I., Oberhofer, M., van Run, P., Wolfson, D., *Enterprise master data management: An SOA approach to managing core information*. Upper Saddle River, NJ: IBM Press/Pearson plc, 2008.
[24] M. Ofner, K. Straub, B. Otto, and H. Oesterle, “Management of the Master Data Lifecycle: A Framework for Analysis,” *Journal of Ent Info Management*, vol. 26, no. 4, pp. 472–491, 2013, doi: 10.1108/JEIM-05-2013-0026.
[25] D. Loshin, *Master data management*. Amsterdam: Elsevier Morgan Kaufmann, 2010.
[26] K. D. Bailey, *Typologies and taxonomies: An introduction to classification techniques*. Thousand Oaks: Sage, 1994. Accessed: Mar. 21 2021.
[27] R. C. Nickerson, J. Muntermann, and U. Varshney, “Taxonomy Development in Information Systems: A Literature Survey and Problem Statement,” in *AMCIS Proceedings*, Lima, Peru, 2010. Accessed: May 3 2021.
[28] R. Nickerson, J. Muntermann, U. Varshney, and H. Isaac, “Taxonomy Development in Information Systems: Developing a Tayxonomy of Mobile Applications,” 2009. Accessed: May 3 2021. [Online]. Available: https://halshs.archives-ouvertes.fr/halshs-00375103
[29] M. Chisholm, *Managing reference data in enterprise databases: Binding corporate data to the wider world*. San Francisco, CA: Kaufmann Publishers, 2001. [Online]. Available: http://www.loc.gov/catdir/description/els031/00043560.html
[30] M. Chisholm, *Master Data versus Reference Data.* [Online]. Available: http://www.dmreview.com/article_sub_articleId_1051002.html (accessed: Jul. 3 2020).